Method and system for reading data

The invention relates to a system and a method for reading data from a document, and to a method of acquiring data.

Nowadays, a multitude of data is stored in an electronic form. This comprises, on the one hand, storage in databases for targeted queries of one or more data sets by means of a computer. On the other hand, this also comprises documents which are provided for retrieval and observation by human users, such as HTML or XML files, tables, structured texts or work pages of a table computation. Similarly as the above-mentioned databases, the latter electronic documents are, however, also computer-readable. For querying single data from such a document (for example, for reading an individual entry in a table) there is, however, no special query interface. The automatic reading of data from such documents which – in contrast to databases – will herein be referred to as weakly structured data, usually requires the provision of an extract instruction in an appropriate computer language, for example, a PERL script or a regular expression which is interpreted by known programs as grep, sed or awk. The provision of such an instruction requires programming expertise and is not very comfortable for a user.

The aim of reading data from a document is usually not only a one-time operation of reading data. However, data changing with time, i.e. updated data are preferably read frequently and with time intervals from such documents. For example, a document (for example, a HTML page) indicated by a given address (URL) in a computer network may comprise an updated table with weather data from different cities. The indicated data, for example, the temperature, will change from day to day. Under circumstances, it is even possible that the absolute location of a range changes by reformatting. For example, one day, the temperature value of Paris may be mentioned in the third row in column 2, and another day in the second column of another row.

The particularly interesting class of documents in which the information may change with time will hereinafter be referred to as "volatile".

It is an object of the invention to provide a system and a method for reading data from a document and a method of acquiring, with which a user can simply create an

extract instruction with which data, particularly from weakly structured volatile documents, can be queried.

This object is solved by a system as defined in claim 1, a method of reading data as defined in claim 9 and a method of detecting data as defined in claim 10. Dependent claims are defined by advantageous embodiments of the invention.

In the solution according to the invention, the user fixes the range of interesting data in a document by means of an input processed by a computer by means of a program running on this computer. By means of the program, an extract instruction is automatically generated for this purpose.

The computer has access to at least one document. The computer is preferably connected to a computer network, for example, the Internet and accesses a remote document via the computer network.

The program running on the computer indicates, for example, the document and invites the user to fix the interesting data range, for example, by marking it with an indicator unit (such as a mouse). Optionally, the user may additionally provide a second input with which one or more further ranges, here referred to as structural ranges, are fixed, which may be helpful in finding the desired range. For example, these structural ranges may be row or column headers which, in a table, lead to a cell with the desired contents.

According to the invention, an extract instruction is automatically generated on the basis of the inputs by the user. The extract instruction is supplied in a form which can be read by a computer and executed by an appropriate interpreter program, and is preferably stored. When executing the extract instruction, the fixed data range of the document is read. When the user has additionally predetermined a structural range, the extract instruction created preferably comprises an address indication corresponding to the position or the content of the predetermined structural range.

In accordance with a further embodiment of the invention, a special grammar is predetermined for the extract instruction. A valid expression is composed from a predetermined sequence of terminal symbols. The grammar preferably comprises address indications with which given positions of a document can be addressed, on the one hand, absolutely (for example, introduction to the document) and, on the other hand, also relatively to a previously indicated range (for example, two rows lower).

The grammar used preferably has a simple structure. The grammar is preferably adapted to the type of the interesting document. For example, a special grammar may be provided for addressing in running texts which can then provide, for example,

addressing on a word and sentence basis (for example, second word in third sentence). Alternatively, a special grammar may be provided for tables with which addressing on the basis of rows and columns is very well possible (for example, third field in the row starting with "Paris").

In accordance with a further embodiment of the invention, the extract instruction is automatically created in that a plurality of valid extract instructions of the predetermined grammar is generated, which extract instructions are checked on whether the interesting data range of the document is read upon their execution. One of the successful extract instructions is selected, for example, with reference to a complexity criterion.

The provision of an automatic extract instruction is preferably not only realized by means of a document, but a plurality of training documents is processed. In this way, there may be a greater probability that the automatically created extract instruction always supplies the desired data, also in the case of volatile documents, without constant adaptation being necessary.

An extract instruction once created and preferably verified with a plurality of training documents is preferably stored. It can then be iterated with time intervals so as to read the current value from the addressed range of a constantly updated document. The value can be further processed in many different ways. For example, current information of different documents that can be called from a computer network can be combined and processed to information compiled in accordance with personal preferences.

These and other aspects of the invention are apparent from and will be elucidated with reference to the embodiments described hereinafter.

In the drawing:

Fig. 1 shows a graph for deriving an extract instruction.

In the embodiment, current weather reports are to be inserted in an automatically provided personal radio program as described in, for example, WO 99/39466. The required current weather information is constantly available from different Internet pages (HTML documents). The user should be allowed to fix the information by means of a computer in a simple manner, which information is subsequently inserted into his daily personal radio program (by means of speech synthesis).

To this end, a computer system with input means (for example, keyboard, mouse) and output means (for example, monitor) is used. The computer is connected to the Internet. A program is installed on the computer, with which program the user can simply formulate an extract instruction for the data that are interesting to him and transmit these data to the service provider who compiles the personal radio program for him. The function of this program will hereinafter be described in detail.

A grammar is provided for the query to be formulated. It can be defined arbitrarily. Such a grammar comprises terminal symbols of the following types:

1. Absolute addressing so as to address an absolutely fixed range within a document (for example, TOP, BOTTOM, ROOT).

2. Relative addressing so as to address locations or ranges within a document, starting from an original location or original range (for example, next_paragraph, previews_word, next_list_item, cell_up, to_first_row, parent_node, first_child, next_sibling).

3. Search commands so as to address locations under given conditions. A search command consists of a search range (for example, in_paragraph, in_sub-tree, within_column) and a condition (for example, contains_text (T), has_format (F), is_a_number, is_smaller_than (n), carries_xml_tag (T)), possibly a relative path to the location where the condition should be relevant (for example, the relative addressing as above under 2.) and an indicator with which, in the case of a plurality of hits, a single one can be selected (for example, first_occurrence, last_occurrence, nth_occurrence (n)).

Extract instructions for different documents can be built up by means of a combination of a plurality of the above-mentioned terminal symbols. For example, it will be evident to those skilled in the art that an extract instruction of the following type can be built up: "Take the third table of the document and select the first occurrence of a number followed by a $ sign in a cell which is present in the row whose first column input is "Canada". Such an extract instruction could be formed, for example, as follows:

```
TOP to_next_table to_next_table to_next_table
find (in_table, is_a_number and has_format ("$")
  and (to_first_column contains_text ("Canada")),
  first_occurrence)
```

The program running on the computer receives inputs from the user, with which the interesting data in a document are indicated. The program then automatically

creates an extract instruction. The extract instruction is formulated in the predetermined grammar. In the case of a corresponding execution, i.e. through a corresponding interpreter, in the relevant document, it supplies the indicated data.

For example, the user lives in Frankfurt and, within the framework of his personal radio program, he would like to be informed daily about the current meteorological values such as temperature and humidity in that city. He searches a HTML page which can be called from the Internet, which page daily gives current information about these topics. The following Table shows, by way of example, the contents of such a page:

Location	Temperature (°C)	Humidity (%)	Clouds (%)
Aachen	24	90	80
Berlin	18	70	30
Frankfurt a.M.	22	60	20
Köln	23	50	95

To insert the information about the current humidity in the personal radio program, the user queries the computer with the program, which query is transmitted to the service provider for the purpose of composing the personal radio program. When executing the program, the user calls the document with the above-mentioned Table. He marks the interesting value, here the value for the humidity in Frankfurt (60), the mark has been underlined in the Table by means of the mouse. Additionally, the user marks the row header ("Frankfurt") as a structural range which can be used when addressing the interesting value.

From this information, the program automatically creates an extract instruction in accordance with the predetermined grammar. The program mode has been given as a pseudo code in the following survey:

1. SET TargetExpression := <empty>
   SET DocumentsAndMarksList := <empty>
2. FOR d IN {all training documents} DO
3. IF TargetExpression is an extract instruction leading to your valid input in the training document d,
   THEN indicate the range marked by TargetExpression
   ask user whether the marked region corresponds to the desired data
   IF user answers with „yes", THEN GOTO5

4. request the user for an input with which the desired range is marked in the training document d. Optionally, the user can also provide one or more marks in structural ranges to be taken into account for the query (if he does not do this, SET A : = <empty>)
5. attach the triplet (d, M, A) to the DocumentsAndMarksList.
6. FOR all extract instructions L which can be derived from the grammar G and do not exceed a predetermined complexity measure.
7. SET count := 0
8. FOR all triplets (t.A, t.M, t.D) IN DocumentsAndMarksList DO
9. IF (MARKING_DUE_TO_LOCATOR_EXPR(t.D, L) == t.M)
AND (t.A $\subseteq$ LOCATOR_EXPRESSION_PATH (t.D, L))
THEN count++
10. DONE (process next triplet in step 8)
11. IF (count> bestcount)
OR ((count == bestcount) AND (COMPLEXITY(L) < COMPLEXITY(bestL))
THEN
SET bestL := L ; SET bestcount := count
12. DONE (continue with the next extract instruction in step 6)
13. DONE (continue with the next document in step 2)
14. RETURN bestL

This program uses the following functions:

MARKING_DUE_TO_LOCATOR-EXPR(Document d, extract instruction):
This function interprets the extract instruction and returns the data of the document d which are at the location marked by the extract instruction.

LOCATOR_EXPRESSION_PATH (Document d, extract instruction):
This function returns a set of ranges which are run through when the extract instruction in document d is executed.

COMPLEXITY(extract instruction L):

A complexity measure for the extract instruction L, for example, the length of the expression. This complexity measure is used for selection when otherwise there are several equivalent extract instructions.

The program operates with a number of training documents. These are preferably different documents that can be called under the same URL at different instants so that the volatility of the interesting document is covered as satisfactorily as possible by the available set of training documents. However, the program may also be used when only few training documents or even a single training document is available. The outer loop (2. – 13.) is then run through a correspondingly smaller number of times.

The program represented as pseudo-code above operates as follows:

An outer loop (2. – 13.) is run through for all available training documents. When an extract instruction TargetExpression is already created, which leads to a valid input, this range is marked and the user is asked whether this is the desired range (step 3). Otherwise, the user is requested to mark the desired range himself (and optionally also one or more structural ranges) (step 4).

The triplet consisting of training documents, desired ranges and (optionally) structural ranges are attached to a list of DocumentsAndMarksList (step 5).

In step 6, a number of extract instructions L is generated from the grammar G. These are preferably all the valid expressions in the grammar G which do not exceed a predetermined complexity measure (for example, the overall length of the expression). It will be easily possible for those skilled in the art to automatically generate valid expressions from the grammar definition.

For each generated extract instruction L, all the available triplets (documents with target ranges and possibly structural ranges) are checked on whether the expression leads to the desired result. If this is the case, a counter is up-counted (steps 8 – 10).

The result of an extract instruction, i.e. the number of correct markings in the available training documents, is compared with the currently highest result (best count). At a higher value, the current expression is maintained as the best candidate. At equal values with the currently best candidate, the expression with the lowest complexity is maintained (step 11).

At the end, the best expression determined in this manner is returned as the extract instruction that has been found (step 14).

The program run will hereinafter be illustrated by way of a simple example.

A simple grammar which may be used, for example, for table structures will be given below by way of example. Terminal symbols are indicated in small case, non-terminal symbols are indicated in capitals:

EXPRESSION ::= top_left_cell ROW_HEADER_SEARCH_EXPR

ROW_HEADER_SEARCH_EXP ::= find (within_column, contains_text(#))
ROW_ELEMENT_SELECTION

ROW_ELEMENT_SELECTION ::= select_entire_cell |
cell_left ROW_ELEMENT_SELECTION

When the above algorithm with this grammar is used for the Table above, in which the number 60 is marked as a target range and the word "Frankfurt" is marked as a structural range, the created extract instruction

TOP find (within_column, contains_text ("Frankfurt"))
cell_right cell_right select_entire_cell

as shown in Fig. 1 could be derived from the grammar. The content "Frankfurt" of the marked structural range is then converted into a "find" indication with which the word "Frankfurt" is searched in the first column (the sign "#" in the grammar is replaced by the content of the selected structural range "Frankfurt").

Starting from the found cell, the searched cell with the content "60" is two cells further to the right, i.e. it is reached by two calls from cell_right. The cell thus found is marked as a whole and supplies the desired content "60".